EXHIBIT D9

SOFTWARE QUALITY ASSURANCE PLAN (SQAP) TEMPLATE

30 September 1997
Version 1.2

FOREWORD

This document is a template of a Software Quality Assurance (SQA) Plan using the guidelines provided in IEEE 730-1989, IEEE Standard for Software Quality Assurance Plans, and IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning. This template should be supplemented with project-specific information to produce an SQA Plan (SQAP) that accurately describes the project's SQA organization, tasks, role, and responsibilities. The planning and documenting of SQA activities must agree and be consistent with the project's Software Development Plan (SDP) and any other project planning document. Additionally, the SQAP must comply with SPAWARSYSCEN SAN DIEGO CA SQA Policy. The SQA Policy is SPAWARSYSCEN SAN DIEGO CA written organizational policy for implementing SQA to provide management with appropriate visibility into the process being used by the software project and of the products being built.

This document supplements the SQA Process document. Refer to Section 4, Create/Maintain SQAP, of the SQA Process document for a description on the use of this template.

SEPO will maintain this SQAP template. As a user of this document report deficiencies and or corrections using the Document Change Request (DCR) found on the next page. SEPO will collect and process this data as inputs for process improvements to the SQAP template.

SQAP Template
Version 1.2
9/30/97

DOCUMENT CHANGE REQUEST (DCR)

Document Title:	Tracking Number:
Name of Submitting Organization:	
Organization Contact:	Phone:
Mailing Address:	
Short Title:	Date:
Change Location: (use section #, figure #, table #, etc.)	
Proposed change:	
Rational for Change:	

Note: For the Software Engineering Process Office (SEPO) to take appropriate action on a change request, please provide a clear description of the recommended change along with supporting rationale.

Send to: Commanding Officer, Space and Naval Warfare Systems Center, D13, 53560 Hull Street, San Diego, CA 92152-5001 or Fax to: (619)553-6249 or Email to: sepo.spawar.navy.mil

DCR Form 9/1997

Administrative Information

Software Engineering Process Office, D13
Space and Naval Warfare Systems Center
53560 Hull Street, San Diego, CA 92152-5001

SOFTWARE QUALITY ASSURANCE (SQA) PLAN TEMPLATE
Version 1.2

SEPO assumes responsibility for this document and updates it as required to meet the needs of users within SPAWARSYSCEN SAN DIEGO CA. SEPO welcomes and solicits feedback from users of this document so that future revisions will reflect improvements, based on organizational experience and lessons learned.

I wish to acknowledge the following SPAWARSYSCEN SAN DIEGO CA Marine Air Traffic Control and Landing System (MATCALS) Project personnel who contributed to this version of this document.

Ron Ballard, MATCALS Project Manager
Ron Irwin, MATCALS Software Development
Rich Cassity, MATCALS Independent Verification and Validation

Ann Hess, SEPO, SPAWARSYSCEN SAN DIEGO CA, D13
Voice: (619)553-3351, Fax: (619)553-6249, Internet: hess@spawar.navy.mil

RECORD OF CHANGES

*__A__ - ADDED __M__ - MODIFIED __D__ - DELETED

CHANGE NUMBER	DATE	NUMBER OF FIGURE, TABLE OR PARAGRAPH	__A*__ __M__ __D__	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER

DOCUMENT CONVENTIONS

[Text]	Replace text.
<*text in italics*>	Notes or instructions to the author. Delete in final format.

This document is an Software Quality Assurance Plan (SQAP) template and should be supplemented with project-specific information to produce an SQAP that accurately describes your project SQA organization. Therefore tailor (add, delete, change, or expand) the information provided in this document.

In some cases where information may already be found in another project document, like the Software Development Plan (SDP), refer to that document rather than duplicate the information in the project SQAP.

The next page is the start of the template which begins with a Project SQA cover sheet. Delete this page and preceding pages in the final format of your project SQAP. Don't forget to update the header page to reflect your document configuration identifier for the project SQAP.

[Project Name]
SOFTWARE QUALITY ASSURANCE PLAN (SQAP)

[Date]

[Project Name]
SOFTWARE QUALITY ASSURANCE PLAN (SQAP)

[Date]

SQAP Approvals:

SQA Manager		**[Date]**
Project Manager		**[Date]**
Program Manager		**[Date]**

RECORD OF CHANGES

*__A__ - ADDED __M__ - MODIFIED __D__ - DELETED

CHANGE NUMBER	DATE	NUMBER OF FIGURE, TABLE OR PARAGRAPH	__A*__ __M__ __D__	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. PURPOSE

The purpose of this plan is to define the [project name] Software Quality Assurance (SQA) organization, SQA tasks and responsibilities; provide reference documents and guidelines to perform the SQA activities; provide the standards, practices and conventions used in carrying out SQA activities; and provide the tools, techniques, and methodologies to support SQA activities, and SQA reporting.

1.1 SCOPE

This plan establishes the SQA activities performed during development and maintenance of the [project name].

This plan is written to follow Space and Naval Warfare Systems Center, San Diego CA (SPAWARSYSCEN SAN DIEGO CA) policy for implementing SQA for [project name]. Specifically, this SQAP will show that the SQA function is in place for this project and that the SQA group has a reporting channel to senior management that is independent of the project manager, the project's software engineering group, and software related groups that includes Software Configuration Management (SCM), System and Software Test, and Logistics.

The goal of the SQA program is to ensure that all software and documentation to be delivered meet all technical requirements. The SQA procedures defined herein shall be used to examine all deliverable software and documentation to determine compliance with technical and performance requirements.

Table 1-1 shows the software life cycle activities of the Computer Software Configuration Items (CSCIs) to which this SQAP applies.

<In Table 1-1, add or delete activities that apply to the project's software lifecycle and as specified in the project's SDP.>

Table 1-1. Software Lifecycle Activities

SOFTWARE LIFECYCLE ACTIVITY
Project Planning and Oversight
Software Development Environment
System Requirements Analysis
System Design
Software Requirements Analysis
Software Design

Software Implementation and Unit Testing
Unit Integration and Testing
CSCI Qualification Testing
CSCI/HWCI Integration and Testing
System Qualification Testing
Software Use Preparation
Software Transition Preparation

1.2 IDENTIFICATION

Table 1-2 shows the CSCIs that this plan applies to.

Table 1-2. CSCI Nomenclature/Identification

NOMENCLATURE	ACRONYM	CSCI NUMBER
[CSCI Name]	[Acronym]	[CSCI ID number]
[CSCI Name]	[Acronym]	[CSCI ID number]
[CSCI Name]	[Acronym]	[CSCI ID number]

Listed below is a brief description of each of the CSCIs developed and maintained by [project name].

a. [**CSCI #1]** - [Include a brief description of the CSCI and its purpose].

b. [**CSCI #2]** - [Include a brief description of the CSCI and its purpose].

c. [**CSCI #3]** - [Include a brief description of the CSCI and its purpose].

1.3 SYSTEM OVERVIEW

The [system name] <*Complete the sentence by providing a description of the system and the intended use of the system.*> The system includes [enter the number of subsystems, e.g., 4] subsystem within the system. Figure 1-3 identifies the CSCIs within each subsystem and highlights those to which this SQAP applies.

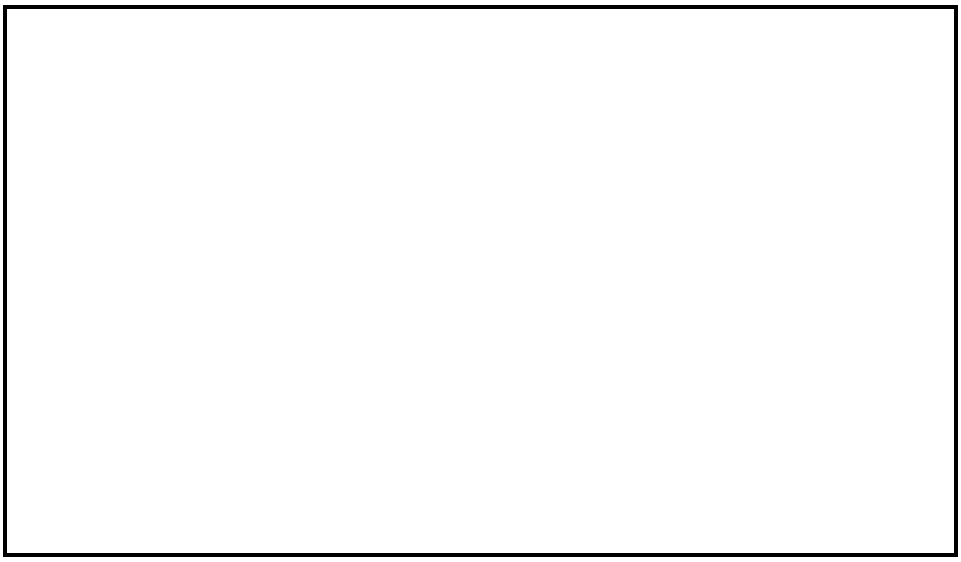

Table 1-3. [System Title] Software

1.4 DOCUMENT OVERVIEW

This document identifies the organizations and procedures to be used to perform activities related to the [project name] software quality assurance program as specified in IEEE Std 730.1-1995.

Section 1 identifies the system to which this SQAP applies; provides an overview of the system and its software functions; summarizes the purpose and contents of the SQAP; and describes the relationship of the SQAP to other management plans.

Section 2 lists all documents referenced in this SQAP.

Section 3 describes each major element of the organization that influences the quality of the software.

Section 4 lists the baseline documents produced and maintained by the project.

Section 5 identifies the standards, practices and conventions.

Section 6 describes the SQA reviews and audits.

Section 7 describes SQA participation in testing.

Section 8 describes problem reporting and corrective action.

Section 9 describes SQA tools, techniques, and methodologies.

Section 10 describes the configuration management tool used for code control.

Section 11 describes SQA's evaluation of media control.

Section 12 describes control of supplier software.

Section 13 describes SQA procedures for records collection, maintenance, and retention.

Section 14 describes SQA training requirements.

Section 15 describes SQA review of the Risk Management process.

Appendix A provides a list of acronyms.

1.5 RELATIONSHIP TO OTHER PLANS

SQA evaluation of the software development processes during all phases of the development and maintenance life cycle is based on the processes defined in the Software Development Plan (SDP) for [project name]. The SDP and its implementation procedures baseline the SQA evaluation criteria. The SQAP is implemented in conjunction with the Software Configuration Management Plan (SCMP).

2. REFERENCE DOCUMENTS

This section lists the documents referenced or used as a source for this SQAP.

<For the following, add or delete documents that were referenced or used as a source for the SQAP.>

2.1 GOVERNMENT DOCUMENTS

a. MIL-STD-498, Software Development and Documentation
b. MIL-STD-973, Configuration Management
c. MIL-STD-1521, Technical Reviews and Audits for Systems, Equipments, and Computer Software
d. SPAWARSYSCEN SAN DIEGO CA Software Quality Assurance Process, Version 1.4, 9/4/97
e. SPAWARSYSCEN SAN DIEGO CA Software Quality Assurance Plan Template, Version 1.2, 8/20/97

2.1.1 PROGRAM DOCUMENTS

a. SPAWARSYSCEN SAN DIEGO CA Software Engineering Process Policy, Draft, SPAWARSYSCEN SAN DIEGO CA INST zzzz.1, 11/1/96
b. SPAWARSYSCEN SAN DIEGO CA Software Quality Assurance Policy, Version 1.0, 12/10/96
c. [Program Title] Program Plan, [Document Date], [Documentation Identification]

2.1.2 PROJECT DOCUMENTS

a. [Project name] Software Development Plan, [Document Date], [Documentation Identification]
b. [Project name] Software Configuration Management Plan, [Document Date], [Documentation Identification]
c. [Project name] Computer Resources Life Cycle Management Plan, [Document Date], [Documentation Identification]

2.2 NON-GOVERNMENT DOCUMENTS

a. IEEE Standard for Software Quality Assurance Plans, IEEE-Std-730-1989, 17 August 1989
b. IEEE Guide for Software Quality Assurance Planning, IEEE-std-730.1-1995, 12 December 1995

c. CMU/SEI-94-HB-01, Carnegie-Mellon University Software Engineering Institute, A Software Process Framework for the SEI Capability Maturity Model (CMM), September 1994
d. CMU/SEI-93-TR-24, Capability Maturity Model for Software, Version 1.1, February 1993
e. IEEE Std 1298/A3563.1, Software Quality Management System

This page intentionally left blank.

3. MANAGEMENT

This section describes each major element of the organization that influences the quality of the software.

3.1 ORGANIZATION

Good software practice call for a measure of independence for the SQA group (such independence is cited as an example of a SQA requirement by the CMM). This independence provides a key strength to SQA; that is, SQA has the freedom, if the quality of the product is being jeopardized, to report this possibility directly above the level of the project. While in practice this rarely occurs (for almost all problems are correctly addressed at the project level), the fact that the SQA group can go above the project level gives it the ability to keep many of these problems at the project level.

Figure 3-1 shows the SQA organization with relation to the project organization.

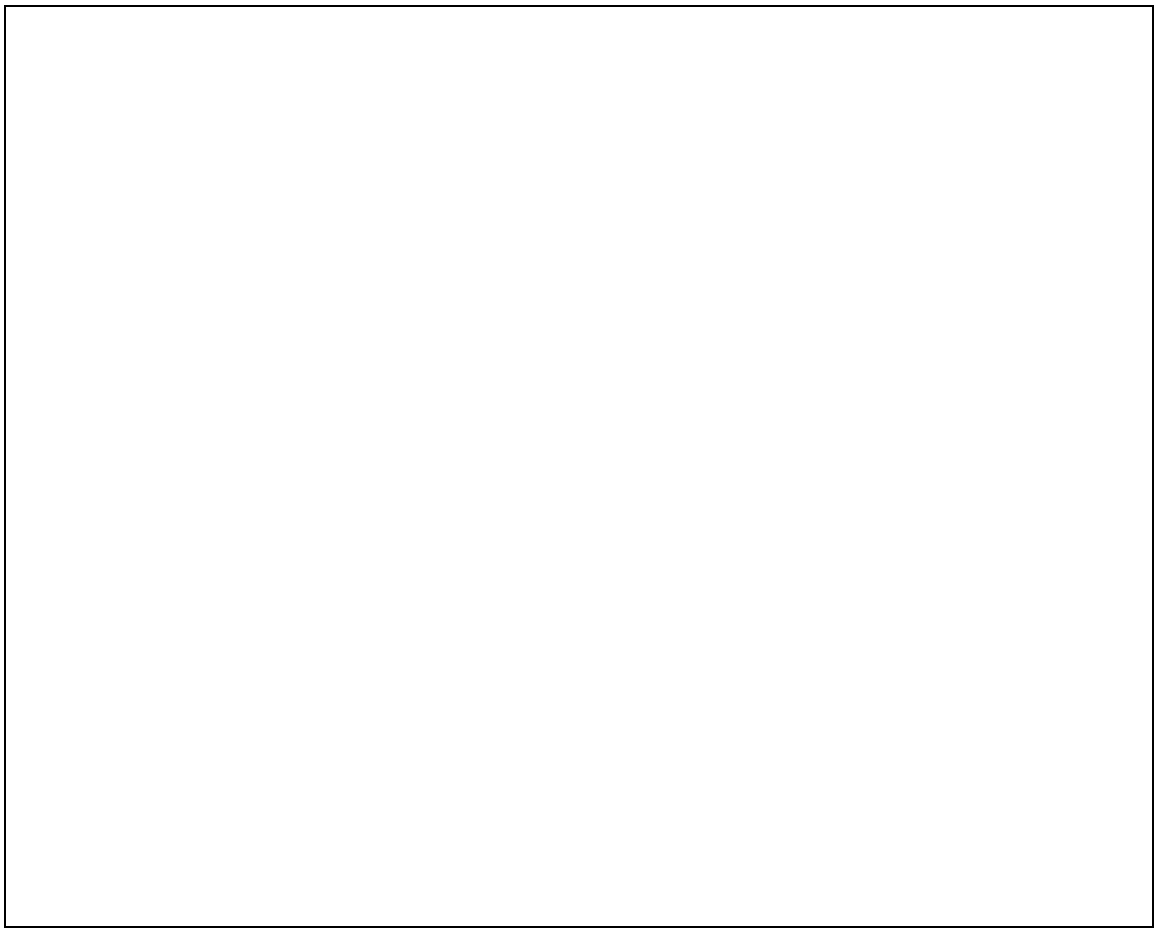

Figure 3-1. [Project Name] Organization

<Replace Figure 3-1 with your project's organizational structure and provide a description of the functional responsibilities for each functional group in the organizational structure.>

<In describing the functional responsibilities, provide the following,
who interacts with SQA
who has authority and delegates responsibilities of interacting functions
reporting relationships among the interacting elements identifying independence/dependence
who has product release authority
who approves the SQAP
reporting lines for escalating conflicts and the method by which conflicts are to be resolved among the elements.

In each case add or delete the functional responsibilities that apply.>

SQA is responsible for ensuring compliance with SQA requirements as delineated in this SQAP. The SQA organization assures the quality of deliverable software and its documentation, non-deliverable software, and the engineering processes used to produce software.

The following describes the functional groups that influence and control software quality.

a. Program Management (Sponsor) is responsible for:
(1) Establishing a quality program by committing the project to implement the SPAWARSYSCEN SAN DIEGO CA Software Engineering Process Policy and the SQA Policy.
(2) Reviewing and approving the [project name] SQAP.
(3) Resolving and following-up on any quality issues raised by SQA.
(4) Identifying an individual or group independent from the project to audit and report on the project's SQA function.
(5) Identifying the quality factors to be implemented in the system and software.
(6) *<fill-in additional functional responsibilities>*.

b. Project Management is responsible for:
(1) Implementing the quality program in accordance with the SPAWARSYSCEN SAN DIEGO CA Software Engineering Process Policy and the SQA Policy.
(2) Identifying the SQA activities to be performed by SQA.
(3) Reviewing and approving the [project name] SQAP.
(4) Identifying and funding an individual or group independent from the project to perform the SQA functions.
(5) Resolving and following-up on any quality issues raised by SQA.
(6) Identifying and ensuring the quality factors to be implemented in the system and software.

(7) Identifying, developing and maintaining planning documents such as the Program Management Plan, SDP, SCMP, Test Plans, and this SQAP.
(8) <*fill-in additional functional responsibilities*>.

c. System Engineering is responsible for:
(1) Reviewing and commenting on the [project name] SQAP.
(2) Implementing the quality program in accordance with this SQAP.
(3) Resolving and following-up on any quality issues raised by SQA related to software engineering activities.
(4) Identifying, implementing, and evaluating the quality factors to be implemented in the system (software and hardware).
(5) Implementing the software engineering practices, processes, and procedures as defined in the [project name] SDP and other program/project planning documents.
(6) <*fill-in additional functional responsibilities*>.

d. Software Design/Development is responsible for:
(1) Reviewing and commenting on the [project name] SQAP.
(2) Implementing the quality program in accordance with this SQAP.
(3) Resolving and following-up on any quality issues raised by SQA related to software design and development.
(4) Identifying, implementing, and evaluating the quality factors to be implemented in the software.
(5) Implementing the software design/development practices, processes, and procedures as defined in the [project name] SDP and other program/project planning documents.
(6) <*fill-in additional functional responsibilities*>.

e. Software Test is responsible for:
(1) Reviewing and commenting on the [project name] SQAP.
(2) Implementing the quality program in accordance with this SQAP.
(3) Resolving and following-up on any quality issues raised by SQA related to software test.
(4) Verifying the quality factors are implemented in the system, specifically software.
(5) Implementing the software test practices, processes, and procedures as defined in the [project name] SDP and other program/project planning documents.
(6) <*fill-in additional functional responsibilities*>.

f. System Test is responsible for:
(1) Reviewing and commenting on the [project name] SQAP.
(2) Implementing the quality program in accordance with this SQAP.
(3) Resolving and following-up on any quality issues raised by SQA as related to system test.
(4) Verifying the quality factors are implemented in the system (software and hardware).

(5) Implementing the system test practices, processes, and procedures as defined in the [project name] SDP and other program/project planning documents.
(6) <*fill-in additional functional responsibilities*>.

g. Logistics is responsible for:
(1) Reviewing and commenting on the [project name] SQAP.
(2) Implementing the quality program in accordance with this SQAP.
(3) <*fill-in additional functional responsibilities*>.

h. Software Configuration Management (SCM) is responsible for:
(1) Reviewing and commenting on the [project name] SQAP.
(2) Implementing the quality program in accordance with this SQAP.
(3) Resolving and following-up on any quality issues raised by SQA related to software CM.
(4) Ensuring the quality factors are implemented in the software related to software CM.
(5) Implementing the software CM practices, processes, and procedures as defined in the [project name] SDP and other program/project planning documents.
(6) <*fill-in additional functional responsibilities*>.

i. Independent Verification and Validation (IV&V) is responsible for:
(1) Reviewing and commenting on the [project name] SQAP.
(2) Implementing the quality program in accordance with the [project name] SQAP.
(3) Resolving and following-up on any quality issues raised by SQA.
(4) Verifying the quality factors are implemented in the system (hardware and software).
(5) Implementing the practices, processes, and procedures as defined for IV&V in the [project name] SDP and other program/project planning documents.
(6) <*fill-in additional functional responsibilities*>.

j. Software Engineering Process Office (SEPO) is responsible for:
(1) Keeping the SPAWARSYSCEN SAN DIEGO CA Software Engineering Process Policy and SQA Policy current.
(2) Maintaining the SQA process document and SQAP template.
(3) Ensuring SQA training availability, either by vendor or SEPO.
(4) Providing assistance in software process engineering and software process improvement.

3.2 RESOURCES

3.2.1 FACILITIES AND EQUIPMENT

SQA will have access to the facilities and equipment as described in the [project name] SDP. SQA will have access to computer resources to perform SQA functions such as process and product evaluations and audits.

3.2.2 PERSONNEL

The SQA effort for this project is [person-year effort or indicate the amount of effort if it is less than 100%].

<Identify the qualifications requirements of the SQA Manager>

The SQA Manager will be familiar with and will be able to apply the standards and guidelines listed in Section 2, Reference Documents. Additionally, the SQA Manager will be familiar with software quality, software development related activities, and structured analysis, design, coding, and testing.

3.3 SQA TASKS

<Describe the portion of the software life cycle covered by this SQAP, the tasks to be performed with special emphasis on SQA activities, and relationship between these tasks and the planned major check-points. The sequence of the tasks should be indicated.>

The scheduling of SQA tasks is driven by the software development schedule. Therefore, an SQA task is performed in relationship to what software development activities are taking place. One or more SQA tasks can be performed concurrently until a task is completed. A task is considered completed when the required report e.g., SQA Reports, Process Audits Reports, etc. are satisfactory completed or action items have been closed. The following tasks, requiring coordination and cooperation with the project team, shall be performed by SQA.

3.3.1 Task: Review Software Products

The [project name] SDP identifies all software products to be developed and evaluated and includes the standards or guidelines to be followed. As required, SQA shall assist the project in identifying the standards or guidelines to be followed in developing the software product. Section 4, Documentation, lists the software products to be evaluated by SQA and describes the review process to be followed.

3.3.2 Task: Evaluate Software Tools

SQA shall conduct evaluations of tools, both existing and planned, used for software development and support. The tools are evaluated for adequacy by assessing whether

they perform the functions for which the tools are intended and for applicability by assessing whether the tools capabilities are needed for the software development or support. Planned tools are evaluated for feasibility by assessing whether they can be developed with the techniques and computer resources available or by procurement. Section 8, Problem Reporting and Corrective Action, provides the format for reporting the results of a software tool evaluation.

3.3.3 Task: Evaluate Facilities

SQA shall evaluate facilities, both existing and planned, for adequacy by assessing whether they provide the needed equipment and space used for software development and support. Section 8, Problem Reporting and Corrective Action, provides the format for reporting the results of evaluating the project's facilities.

3.3.4 Task: Evaluate Software Products Review Process

This SQA task assures that all software products, which may include representations of information other than traditional hard-copy documents, exist and have undergone software product evaluation, testing, and corrective action as required by the standard.

SQA shall check that software products that are ready for review are reviewed, ensure results are reported and issues or problems reported are resolved in accordance with the project's SDP and procedures.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be process in accordance with the Corrective Action Process.

3.3.5 Task: Evaluate Project Planning and Oversight Process

Project planning and oversight involves project management to develop and document plans for Software Development, CSCI and System Test, Software Installation, and Software Transition. Section 2 lists the program/project plans. For project documents to be developed, SQA will assist in identifying the appropriate guidelines, standards, or Data Item Description (DIDs), and will assist with the tailoring of those guidelines, standards, or DIDs to meet the project's needs.

SQA shall evaluate that the project conducts the relevant activities stated in the Program and Project plans. To ensure that these activities are performed as planned, SQA will audit the processes that define the activity, and will use the SDP or planning document as the measure of whether those activities are being met.

The results of this task shall be documented using the Process Audit Form described in Section 8. Any recommended changes to those plans will require update and approval by project management in accordance with the configuration management procedure as described in the [project name] SCMP.

3.3.6 Task: Evaluate System Requirements Analysis Process

Requirements analysis establishes a common understanding of the customer's requirements between customer and the software project team. An agreement with the customer on the requirements for the software project is established and maintained. This agreement is known as allocating system requirements to software and hardware. Section 4, Documentation, list the system requirements documents.

SQA shall perform the following:

a. Ensure that the correct participants are involved in the requirements definition and allocation process to identify all user needs.

b. Ensure that requirements are reviewed to determine if they are feasible to implement, clearly stated, and consistent.

c. Ensure that changes to allocated requirements, work products and activities are identified, reviewed, and tracked to closure.

d. Ensure that project personnel involved in the requirements definition and allocation process are trained in the necessary procedures and standards applicable to their area of responsibility in order to do the job correctly.

e. Ensure that the commitments resulting from allocated requirements are negotiated and agreed upon by the affected groups.

f. Verify that commitments are documented, communicated, reviewed, and accepted.

g. Ensure that allocated requirements identified as having potential problems are reviewed with the group responsible for analyzing system requirements and documents, and that necessary changes are made.

h. Verify that the prescribed processes for defining, documenting, and allocating requirements are followed and documented.

i. Confirm that a CM process is in place to control and manage the baseline.

j. Verify that requirements are documented, managed, controlled, and traced (preferably via a matrix).

k. Verify that the agreed upon requirements are addressed in the SDP.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.7 Task: Evaluate System Design Process

System-wide design decisions are decisions about the system's behavioral design and other decisions affecting the selection and design of system components. System architectural design is organizing a system into subsystems, organizing a subsystem into Hardware Configuration Items (HWCIs), Computer Software Configuration Items (CSCIs), and manual operations, or other variations as appropriate. Section 4, Documentation, lists the system design documents.

SQA shall perform the following:

a. Ensure that lifecycle documents and the traceability matrix are prepared and kept current and consistent.

b. Verify that relevant lifecycle documents are updated and based on approved requirements change.

c. Ensure that design walkthroughs (peer reviews) evaluate compliance of the design to the requirements, identify defects in the design, and evaluate and report design alternatives.

d. Participate in a sampled set of design walkthroughs and verify all walkthroughs are conducted.

e. Identify defects, verify resolution for previous identified defects, and ensure change control integrity.

f. Selectively review and audit the content of system design documents.

g. Identify lack of compliance with standards and determine corrective actions.

h. Determine whether the requirements and accompanying design and tools conform to standards, and whether waivers are needed prior to continuing software development.

i. Review demonstration prototypes for compliance with requirements and standards.

j. Ensure that the demonstration conforms to standards and procedures.

k. Review the status of design milestones.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.8 Task: Evaluate Software Requirements Analysis Process

The purpose o software requirements analysis is to formulate, document and manage the software requirements baseline; respond to requests for clarification, correction or change; analyze impacts; revise the software requirements specification; and manage the software requirements analysis and change process. Section 4, Documentation, lists the software requirements documents.

SQA shall perform the following:

a. Ensure that the software requirements definition and analysis process and associated requirements reviews are conducted in accordance with the standards and procedures established by the project and as described in the SDP.

b. Ensure that action items resulting from reviews of the software requirements analysis are resolved in accordance with these standards and procedures.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.9 Task: Evaluate Software Design Process

Preliminary design activity determines the overall structure of the software to be built. Based on the requirements identified in the previous phase, the software is partitioned into modules, and the function(s) of each module and relationships among these modules are defined.

A goal of detailed design is to define logically how the software will satisfy the allocated requirements. The level of detail of this design must be such that the coding of the computer program can be accomplished by someone other than the original designer. Section 4, Documentation, lists the software design documents.

SQA shall perform the following:

a. Ensure that the software design process and associated design reviews are conducted in accordance with standards and procedures established by the project and as described in the SDP.

b. Ensure that action items resulting from reviews of the design are resolved in accordance with these standards and procedures.

c. Evaluate the method used for tracking and documenting the development of a software unit in order to determine the method's utility as a management tool for assessing software unit development progress. Example criteria to be applied for the evaluation are the inclusion of schedule information, results of audits, and an indication of internal review and approval of its constituent parts.

d. Ensure that the method, such as the Software Development File (SDF) or Unit Development folder (UDF), used for tracking and documenting the development of a software unit is implemented and is kept current.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.10 Task: Evaluate Coding and Unit Testing Process

Software implementation or coding is the point in the software development cycle at which the design is finally implemented. The process includes unit testing of the software code.

SQA shall perform the following:

a. Ensure that the coding process, associated code reviews, and software unit testing are conducted in conformance with the standards and procedures established by the project and as described in the SDP.

b. Ensure that action items resulting from reviews of the code are resolved in accordance with these standards and procedures.

c. Ensure that the SDF used for tracking and documenting the development of a software unit is implemented and is kept current.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective

action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.11 Task: Evaluate Unit Integration and Testing, CSCI Qualification Testing, CSCI/HWCI Integration and Testing, and System Qualification Testing

Software integration and test activities combine individually developed components together in the developing environment to ensure that they work together to complete the software and system functionality. For joint hardware and software development, integration requires close synchronization of hardware and software to meet designated integration and test milestones.

In the integration and test phase of the development lifecycle, the testing focus shifts from an individual component correctness to the proper operation of interfaces between components, the flow of information through the system, and the satisfaction of system requirements.

SQA shall perform the following:

a. Ensure that software test activities are identified, test environments have been defined, and guidelines for testing have been designed. SQA will verify the software integration process, software integration testing activities and the software performance testing activities are being performed in accordance with the SDP, the software design, the plan for software testing, and established software standards and procedures.

b. Ensure any transfer of control of code to personnel performing software integration testing or software performance testing is being accomplished in accordance with established software standards and procedures.

c. Ensure as many of the software integration tests as necessary and all of the software performance tests are witnessed to ensure that the approved test procedures are being followed, that accurate records of test results are being kept, that all discrepancies discovered during the tests are being properly reported, that test results are being analyzed, and the associated test reports are completed.

d. Verify that discrepancies discovered during software integration and performance tests are identified, analyzed, and corrected; software unit tests, and software integration tests are re-executed as necessary to validate corrections made to the code; and the software unit's design, code, and test is updated based on the results of software integration testing, software performance testing, and corrective action process.

e. Ensure the software performance tests produce results that will permit determination of performance parameters of the software.

f. Ensure that the responsibility for testing and for reporting on results has been assigned to a specific organizational element

g. Ensure that procedures are established for monitoring informal testing

h. Review the Software Test Plan and Software Test Descriptions for compliance with requirements and standards.

i. Ensure that the software is tested

j. Monitor test activities, witness tests, and certify test results

k. Ensure that requirements have been established for the certification or calibration of all support software or hardware used during tests.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.12 Task: Evaluate End-item delivery

This activity is applicable for those projects providing a one-time delivery of a product and may also be interpreted as required deliveries for a specified time period or time frame.

SQA shall evaluate the activities in preparation for end-item delivery to ensure that program or project requirement, if any, for functional and physical audits of the end-item products are being satisfied. In some cases, the SQA organization should be allowed to prohibit delivery of certain items, such as documentation, code, or a system, if the project fails to meet contractual requirements or standards.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.13 Task: Evaluate the Corrective Action Process

The corrective action process describes the steps for (1) problem identification and correction occurring during software development to ensure early detection of actual or potential problems, (2) reporting of the problem to the proper authority, (3) analysis of the problem in order to propose corrective measures, (4) timely and complete correction

action, and (5) the recording and follow-up of each problem's status. Problems in this context include documentation errors, software errors, and noncompliance with standards and procedures.

SQA shall perform the following:

a. Periodically review the corrective action processes and their results against the SCMP in order to assess the effectiveness of the correction action process.

b. Perform periodic analysis of all reported problems to identify trends that may disclose generic problem areas. These analyses shall include the study of the causes, magnitude of impact, frequency of occurrence, and preventive measures.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.14 Task: Media Certification

SQA shall certify that the media containing the source code and the media containing the object code which are delivered to the procuring agency correspond to one another. SQA shall certify also that the software version represented by this media matches that on which software performance testing was performed, or correctly represents an authorized update of the code, as applicable.

SQA reports, together with the corrective action records, software test reports, and software product evaluation records can constitute the required evidence for certification.

3.3.15 Task: Nondeliverable Software Certification

The project may use non-deliverable software in the development of deliverable software as long as the operation and support of the deliverable software after delivery to the acquirer do not depend on the non-deliverable software or provision is made to ensure that the acquirer has or can obtain the same software. SQA shall certify that the use of non-deliverable software meets the above criteria, that is, deliverable software is not dependent on non-deliverable software to execute, or verify that the acquirer can obtain the same software. SQA shall ensure that all non-deliverable software used on the project performs its intended functions.

SQA reports, together with the corrective action records, software test reports, and software product evaluation records can constitute the required evidence for certification.

3.3.16 Task: Evaluate Storage and Handling Process

SQA shall verify that there is an established plan, methodology, or set of procedures for storage and handling of the media. SQA shall evaluate the storage of the software product and documentation to ensure that: storage areas for paper products or media are free from adverse environmental effects such as high humidity, magnetic forces, and dust.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.17 Task: Evaluate Subcontractor Control

SQA shall be responsible for ensuring that the quality of all software products from subcontractors conforms to the contract requirements and that the subcontractor's CM plan and procedures are being followed.

SQA reports, together with the corrective action records, and software product evaluation records shall be provided to project management for corrective action by the subcontractor as required.

3.3.18 Task: Evaluate Deviations and Waivers

SQA shall assist program or project management, with requests for deviations and waivers if required, and verify that the deviation or waiver request is processed in accordance with the project's SCMP and approved by the approving agency.

3.3.19 Task: Evaluate Configuration Management Process

Configuration Management is the discipline that applies technical and administrative direction and surveillance to (1) Identify and document the functional and physical characteristics of CIs, (2) Control the changes to CIs and their related documentation, (3) Record and report information needed to manage CSCIs effectively, including the status of proposed changes and the implementation status of approved changes, and (4) Audit the CIs to verify conformance to specifications, interface control documents, and other contract requirements.

SQA shall evaluate the following:

a. Ensure that configuration identification of documents, code, and computer data have established standards for titling, naming, and describing change status.

b. Ensure that baseline management of changes to the developmental baseline (including documents, code and computer data) are identified, reviewed, implemented, and incorporated in accordance with established procedures.

c. Ensure configuration control of changes to baseline documents and software are being managed in accordance with CM requirements as stated in the SCMP.

d. Ensure configuration status accounting reports are prepared at established times, are prepared in accordance with established procedures, and report the status of items that are significant with respect to the management of the configuration of the software product and documentation.

e. Ensure that the personnel assigned to participate in the configuration audits comply with the SCMP.

f. Ensure for document control that only approved, up-to-date documentation is being used by project personnel, and that the document distribution process results in receipt of correct documentation.

g. Ensure that the program support library is the single place of storage for the baseline version of all software. Ensure that the identification of all software includes the software name and a unique version identifier. The evaluation shall also determine that control of access to software products is being properly exercised and that unauthorized changes to master files cannot occur.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.20 Task: Evaluate Software Development Library Control Process

The SDL functions as the main control point for software CM. A SDL contains all units of code developed for evolving project CSCIs, as well as carefully identified listings, patches, errata, CSCI and system magnetic tapes and disk packs, and job control streams for operating or building software systems. The SDL also contains previous versions of the operational software system in the form of magnetic tapes or disk packs.

SQA shall perform the following:

a. Ensure the establishment of the SDL and procedures to govern its operation.

b. Ensure that documentation and computer program materials are approved and placed under library control.

c. Ensure the establishment of formal release procedures for CM approved documentation and software versions.

d. Ensure that library controls prevent unauthorized changes to the controlled software and ensure the incorporation of all approved changes.

The results of this task shall be documented using the Process Audit Form described in Section 8 and provided to project management. SQA's recommendation for corrective action requires project management's disposition and will be processed in accordance with the Corrective Action Process.

3.3.21 Task: Evaluate non-developmental software

Non-Developmental Software (NDS) is software that is provided by the contractor, the Government, or a third party. NDS may be referred to as reusable software, Government-furnished software, or commercially available software depending on it source. SQA shall ensure that non-developmental software performs its intended functions.

SQA reports, together with the corrective action records, software test reports, and software product evaluation records can constitute the required evidence for certifying the software performs its intended functions.

3.3.22 Task: Perform Configuration Audits

SQA may be required to perform or assist in a formal configuration audit in accordance with the project SCMP. A configuration audit is a formal examination of a CSCI. SQA shall perform or assist in the Function Configuration Audit (FCA) and the Physical Configuration Audit (PCA) as detailed in the SCM Plan.

The results of the FCA and PCA shall be reported using the format prescribed in the SCM Plan.

3.3.23 Task: Verifying Implementation of Requirements Management KPA

The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project.

SQA shall review and/or audit the activities and work products for managing the allocated requirements and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.24 Task: Verifying Implementation of Software Project Planning KPA

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project.

SQA shall review and/or audit the activities and work products for software project planning and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.25 Task: Verifying Implementation of Software Project Tracking and Oversight KPA

The purpose of Software Project Tracking and Oversight is to establish adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans.

SQA shall review and/or audit the activities and work products for software project tracking and oversight and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.26 Task: Verifying Implementation of Software Subcontract Management KPA

The purpose of Software Subcontract Management is to select qualified software subcontractors and manage them effectively.

SQA shall review and/or audit the activities and work products for software subcontract and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.27 Task: Verifying Implementation of Software Configuration Management KPA

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle.

SQA shall review and/or audit the activities and work products for software configuration management and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.28 Task: Verifying Implementation of Organization Process Definition

The purpose of Organization Process Definition is to develop and maintain a usable set of software process assets that improve process performance across the projects and provide a basis for cumulative, long-term benefits to the organization.

SQA shall review and/or audit the organization's activities and work products for developing and maintaining the organization's standard software process and related process assets and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.29 Task: Verifying Implementation of Integrated Software Management

The purpose of Integrated Software Management is to integrate the software engineering and management activities into a coherent, defined software process that is tailored form the organization's standard software process and related process assets, which are described in Organization Process Definition.

SQA shall review and/or audit the activities and work products for managing the software project and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.30 Task: Verifying Implementation of Software Product Engineering

The purpose of Software Product Engineering is to consistently perform a well-defined engineering process that integrates all the software engineering activities to produce correct, consistent software products effectively and efficiently.

SQA shall review and/or audit the activities and work products for software product engineering and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.31 Task: Verifying Implementation of Intergroup Coordination

The purpose of Intergroup Coordination is to establish a means for the software engineering group to participate actively with the other engineering groups so the project is better able to satisfy the customer's needs effectively and efficiently.

SQA shall review and/or audit the activities and work products for intergroup coordination and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.3.32 Task: Verifying Implementation of Peer Reviews

The purpose of Peer Reviews is to remove defects from the software work products early and efficiently.

SQA shall review and/or audit the activities and work products for peer reviews and reports the results. The Capability Maturity Model (CMM), Version 1.1, will be used as the guideline for the review and/or audit.

3.4 RESPONSIBILITIES

<This paragraph should identify the specific organizational elements responsible for each task.>

The ultimate responsibility for the quality of the [project name] software and associated documentation produced by [SPAWARSYSCEN SAN DIEGO CA or Agency Name} rests with the [project name] Software Project Manager. The SQA Manager is responsible to the Software Project Manager and shall implement the SQA procedures defined in this plan.

SQA derives its authority from the Project Manager through the [SPAWARSYSCEN SAN DIEGO CA Branch/Division/Department or Agency Name] Manager. SQA shall monitor project staff activities and review products for compliance to applicable standards, procedures, and the SDP. The results of SQA monitoring and analysis along with SQA's recommendations for corrective action shall be reported to the Software Project Manager, and, as required, to the [SPAWARSYSCEN SAN DIEGO CA Branch/Division/Department or Agency Name] Manager. All documents and software approved by the Software Project Manager for release to [user activity] shall have been reviewed and approved by SQA. Table 3-1 is a responsibility matrix for the tasks identified in Section 3.3, Tasks.

Table 3-1. Responsibility Matrix

SQAP	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Develop/Document	X		X						

SQAP									
Review SQAP	X	X	X	X	X	X	X	X	X
Approve SQAP	X	X	X						

Review Software Products	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Review products	X	X	X	X	X	X	X	X	X
Rework by author	Applies as applicable								
Approve product		X	X						

Evaluate Software Tools	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Evaluate tool	X								
Resolve Audit Findings		X	X						

Evaluate Software Facilities	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Evaluate facilities	X								
Resolve Audit Findings		X	X						

Proj Planning & Oversight (PP&O) Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/Document SDP and other project plans (Test Plan, Training Plan, CRLCMP)		X	X						
Review plans	X	X	X	X	X	X	X	X	X
Approve plans		X	X						
Evaluate PP&O process	X								
Resolve Audit Findings		X	X						

System Requirements Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/document Sys Rqmts					X				X
CM Sys Rqmts				X					
Review Sys Rqmts	X	X	X		X	X	X	X	X
Approve Sys Rqmts		X	X						

Evaluate/report Sys Rqmts Process	X								
Resolve Audit Findings		X	X						

System Design Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/document Sys Design					X				
CM Sys Design				X					
Review Sys Design	X	X	X		X	X	X	X	X
Approve Sys Design		X	X						
Evaluate/report Sys Design process	X								
Resolve Audit Findings		X	X						

Software Requirements Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/document SW Rqmts						X	X		
CM SW Rqmts				X					
Review SW Design	X	X	X		X	X	X	X	X
Approve SW Rqmts		X	X						
Maintain SDL and SDFs				X	X	X			
Evaluate/report SW Rqmts Process	X								
Resolve Audit Findings		X	X						

Software Design Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/document SW Design						X	X		
CM SW Design				X					
Review SW Design	X	X	X		X	X	X	X	X
Approve SW Design		X	X						
Maintain SDL and SDFs				X		X			
Evaluate/report SW Design process	X								
Resolve Audit		X	X						

Findings									

Software Implementation & Unit Testing Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logistics**
Develop/fix code						X			
Code review	X					X	X		
Unit Test						X	X		
Maintain SDL and SDFs				X		X	X		
Maintain STR process				X					
Evaluate/report coding & unit testing process	X								
Resolve Audit Findings		X	X						

Unit Integration and Testing Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Integrate SW						X			
Test Integrated SW							X		
Fix errors						X			
Maintain SDL and SDFs				X		X	X		
Maintain STR process				X					
Evaluate/report SW integration test process	X								
Resolve Audit Findings		X	X						

CSCI Qualification Testing Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Performance Test							X	X	
Fix errors						X			
Maintain SDL and SDFs				X		X	X	X	
Maintain STR process				X					
Evaluate/report SW performance test process	X								
Resolve Audit Findings		X	X						

End-item Delivery Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Prepare/document version release doc				X					
Review version release doc	X				X	X	X	X	X
Approve version release doc			X						
Evaluate/report End-item delivery process	X								
Resolve Audit Findings		X	X						

Corrective Action Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Follow corrective action process	X	X	X	X	X	X	X	X	X
Maintain corrective action process				X					
Evaluate/report correction action process	X								
Resolve Audit Findings		X	X						

Certification (media certif., nondeliv s/w)	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Follow certification process	X			X			X	X	X
Certify s/w	X			X					
Evaluate/report certification process	X								
Resolve Audit Findings		X	X						

Storage & Handling Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Follow storage and handling process	X			X		X	X	X	X
Evaluate/report storage and handling process	X								
Resolve Audit Findings		X	X						

Subcontractor Control	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Evaluate subcontractor software products	X		X	X	X	X	X	X	X

Evaluate/report subcontractor control process	X								
Resolve Audit Findings		X	X						

Deviations & Waivers	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Document deviations & waivers		X	X						
Recommend Approval			X						
Approve		Major	Mino r						
Evaluate/report deviation & waiver process	X								
Resolve Audit Findings		X	X						

Configuration Management Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/Document SCM Plan				X					
Review SCMP	X	X	X		X	X	X	X	X
Approved SCMP		X	X	X					
Follow SCM processes	X	X	X	X	X	X	X	X	X
Document SCM procedures				X					
Evaluate/report SCM process	X								
Resolve Audit Findings		X	X						

SW Development Library Control Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Establish SDL				X					
Follow SDL	X		X	X	X	X	X	X	X

procedures									
Evaluate/report SDL process	X								
Resolve Audit Findings		X	X						

Evaluate Non-Developmental SW	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Evaluate non-development SW	X				X	X	X	X	X
Evaluate/report non-developmental SW process	X								
Resolve Audit Findings		X	X						

Configuration Audits	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Assist/Perform configuration audits	X			X	X	X	X	X	X
Evaluate/report configuration audit process	X								
Resolve Audit Findings		X	X						

Verifying Implementation of KPA	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Requirements Management KPA	X								
Software Project Planning KPA	X								
Software Project Tracking and Oversight KPA	X								
Software Subcontractor Management KPA	X								
Software Configuration	X								

Management KPA									
Organizational Process Definition KPA	X								
Integrated Software Management KPA	X								
Software Product Engineering KPA	X								
Intergroup Coordination KPA	X								
Peer Reviews KPA	X								
Resolve Audit Findings		X	X						

3.5 SCHEDULE

SQA schedules are closely coordinated with the software development schedule in the SDP. Process audits will be performed at the beginning of each new phase of development to verify that the appropriate processes are correctly implemented as defined in the planning documents. In addition, spot checks (unscheduled audits) will be made during each phase of development to verify that the processes and desktop procedures are being followed. Section 3.3 identifies the SQA activities through the software life cycle

4. DOCUMENTATION

The documentation which specifies, describes and supports the [project name] software or the software development process shall be created and updated periodically throughout the development cycle.

Table 4-1 is a list of [project name] software deliverable products and the associated standard or guidelines used to develop and maintain the software products. Any tailoring guidelines are also found in Table 4-1. Table 4-2 is a list of non-deliverable products.

For project's software documents to be developed and not yet listed in Tables 4-1 and 4-2, SQA will assist in identifying the specifications, standards, and Data Item Descriptions (DIDs) to be followed in the preparation of the required documentation.

<List the software products that will be developed/maintained and identify the associated Data Item Description (DID) or standard or guidelines that are used to develop/ maintain the software product to which this SQAP applies in Table 4-1. If there are any tailoring guidelines provide that information in Table 4-1. Identify all nondeliverable products in Table 4-2.>

Table 4-1. Deliverable Software Products

NOMENCLATURE	DELIVERABLE DOCUMENTATION	DID, STANDARD, GUIDELINE
[CSCI Name]	[DOCUMENT TYPE, e.g., SSS]	[DID, e.g., DI-IPSC-81431 of MIL-STD-498]
[CSCI Name]	[DOCUMENT TYPE]	[DID,STANDARD, GUIDELINE]
[CSCI Name]	[DOCUMENT TYPE]	[DID,STANDARD, GUIDELINE]

Table 4-2. Nondeliverable Software Products

DOCUMENT TITLE
[Document title]
[Document title]
[Document title]

<State how the documents are to be checked for adequacy. The document review process should include the criteria and the identification of the review or audit by which the adequacy of each document shall be confirmed.>

All document will undergo a Peer Review (walkthrough or Formal Inspection) in accordance with the Peer Review process described in the [project name] SDP. The following provides an overview of the Peer Review process:

1. Walkthrough *(can also be called Non-Author Review or Technical Review)* - the item being reviewed is presented by the primary author in an informal setting with his or her peers. Defects noted are recorded and the author is obligated to address or fix them.

2. Formal Inspections - the item being reviewed is formally presented and discussed in a group meeting conducted by a facilitator rather than the item's primary author. Errors discovered are rigorously recorded, categorized, and analyzed for trends. The formal inspection is performed in accordance with the SPAWARSYSCEN SAN DIEGO CA Formal Inspection Process.

The following criteria apply to a peer review:

(1) Item completeness - Determine whether the item fully met its intended objectives

(2) Problem Identification - Identify problems as early as possible in order to correct them before they are compounded in subsequent project phases

(3) Compliance with standards - Ensure the item complies with established or require standards, or that waivers are sought where it does not meet them

(4) Risk identification - Identify potential risk areas in the project so risks can be managed and mitigated as the project progresses

(5) Traceability - Ensure the item is traceable possibly through a matrix which will help to verify that the item satisfies its requirements and to provide input to other products being developed or maintained.

A peer review requires the following decision by the peer review attendees: (1) Accept the product without further modification, (2) Reject the product due to severe errors (once corrected, another review must be performed), or (3) Accept the product provisionally (minor errors have been encountered and must be corrected, but no additional review will be required).

Upon completion of a peer review, the software product will be submitted to SCM and placed under CM control. The software product will then be processed in accordance

with the SCM software product approval and release process as described in the [project name] SCMP.

5. STANDARDS, PRACTICES, CONVENTIONS AND METRICS

To ensure the delivery of a fully conforming, high-quality product every individual assigned to the project will participate in quality assurance. The SDP defines the procedures by which the software development staff shall ensure the quality of the product during the development process. The remainder of this section describes the procedures used by SQA to ensure that the quality assurance provisions of this SQAP and applicable standards, practices, conventions, and metrics are met.

<Identify the standards (mandatory requirements) to be applied. State how compliance with these items is to be monitored and assured.>

[MIL-STD-498, 12207, 1498] is the software development standard used by the [project name] and any tailoring of this standard is documented in the SDP. Section 3 identifies SQA's evaluation of the requirements, design, implementation, and test phase to ensure compliance with [MIL-STD-498, 12207, 1498] and the SDP.

Section 4, Documentation, identifies the associated DID for each software product to be developed and maintained. Any tailoring of the DID is described in the SDP. SQA will ensure documentation format and content complies with the DID and the SDP.

Standards for logic structure, coding, and code comments are described in the SDP. SQA will ensure source code complies with these standards as detailed in the SDP.

Standards and practices for testing are described in the SDP. SQA will ensure testing activities complies with the SDP.

5.1 Metrics

<Identify or reference the standards, practices, and conventions to be used in the definition, collection and utilization of software metrics data. Cite any internal (e.g., project, corporate) and external (e.g., user, customer) requirements or standards with which metrics practices must comply. IEEE Std 1045-1992 describes conventions for counting the results of the development processes. IEEE Std 1061-1992 provides a methodology for selecting and implementing process and product metrics. IEEE Std 982.1-1988 and Std 982.2-1988 provide various measures for use in different life cycle phases to gain confidence in the building of reliable software. In order to keep metrics simple, the following cost and schedule metrics are offered.)

The following measurements will be made and used to determine the cost and schedule status of the SQA activities:

SQA milestone dates (planned)
SQA milestone dates (completed)

SQA work scheduled (planned)
SQA work completed (actual)
SQA effort expended (planned)
SQA effort expended (actual)
SQA funds expended (planned)
SQA funds expended (actual)

SQA is responsible for reporting these metrics and providing this metric report to the Project Manager.

6. REVIEWS AND AUDITS

<Define the technical and managerial reviews and audits to be conducted. State how the reviews and audits are to be accomplished. State what further actions are required and how they are to be implemented and verified. The type and scope of technical reviews depends heavily on the size, scope, risk and criticality of the software project. The reviews and audits identified here should be the same as specified in the project SDP. >

Table 6-1 identifies the required reviews and audits for the system and software development phases.

Table 6-1. Reviews and Audits

SYSTEM AND SOFTWARE DEVELOPMENT PHASE	SOFTWARE PRODUCTS	REQUIRED AUDITS AND REVIEWS
System Requirements	(1) SSS, IRS, OCD (2) SDP, SCMP, SQAP	(1) System Requirements Review (SRR) (2) Process Audits (3) Management Review (4) Peer review
System Design	(1) SSDD, IDD	(1) System Design Review (SDR) (2) Process Audits (3) Management Review (4) Peer Review
Software Requirements	(1) SRS, IRS	(1) Software Specification Review (SSR) (2) Process Audits (3) Management Review (4) Peer Review

Software Design	(1) SDD, DBDD, IDD	(1a) Software Preliminary Design Review (PDR) (1b) Software Critical Design Review (CDR) (2) Process Audits (3) Managerial Review (4) Peer Review
Software Implementation	Software products	(1) Process Audits (2) Management Review (3) Peer Review
Test	(1) Test Documentation	(1a) Software Test Readiness Review (TRR) (1b) Formal Qualification Review (FQR) (2) Process Audits (3) Managerial Review (4) Functional Configuration Audit (5) Peer Review
Software Release	(1) SVD, User documentation	(1)Production Readiness Review (PRR) (2) Process Audits (3) Management Review (4) Physical Configuration Audit (5) Peer Review

Note: Peer review was discussed in Section 4, Documentation.

6.1.1 Technical Reviews

A primary component of engineering quality into software is the conduct of technical reviews of software products, both deliverable and nondeliverable. Participants of a technical review shall include persons with technical knowledge of the software products to be reviewed. The purpose of the technical review will be to focus on in-progress and final software products rather than the materials generated especially for the review. SQA

will assure that technical reviews are accomplished and will selectively attend them in accordance with approved sampling techniques. The guidelines of MIL-STD-1521B, Technical Reviews and Audits for Systems, Equipments, and Computer Software, will be followed in conducting a technical review. The following summarizes the technical reviews:

System Requirements Review (SRR) - the objective is to ascertain the adequacy of the developer's efforts in defining system requirements.

System Design Review (SDR) - the objective is to evaluate the optimization, correlation, completeness, and risks associated with the allocated technical requirements. Also included is a summary review of the system engineering process which produced the allocated technical requirements and of the engineering planning for the next phase of effort.

Software Specification Review (SSR) - the objective is to review the finalized Computer Software configuration Item (CSCI) requirements and operational concept. A successful SSR shows that the SRS, IRS, and Operational Concept Document form a satisfactory basis for proceeding into preliminary software design.

Software Preliminary Design Review (PDR) - the objective is to evaluate the progress, consistency, and technical adequacy of the selected top-level design and test approach, compatibility between software requirements and preliminary design, and the preliminary version of the operation and support documents.

Software Critical Design Review (CDR) - the objective is to determine acceptability of the detailed design, performance, and test characteristics of the design solution, and on the adequacy of the operation and support documents.

Software Test Readiness Review (TRR) - the objective is to determine whether the software test procedures are complete and to assure that the developer is prepared for formal CSCI testing.

Formal Qualification Review (FQR) - the test, inspection, or analytical process by which a group of configuration items comprising the system are verified to have met specific program or project management performance requirements.

Production Readiness Review (PRR) - the objective is to determine the status of completion of the specific actions which must be satisfactorily accomplished prior to executing a production go-ahead decision.

Technical reviews will be conducted to review evolving software products, demonstrate proposed technical solutions, and provide insight and obtain feedback on the technical effort. The outcome of a technical review will be to:

1. Surface and resolve technical issues.

2. Review project status, specifically surface near- and long-term risk regarding technical, costs, and schedule issues.

3. Arrive at agreed-upon mitigation strategies for identified risks, within the authority of those present.

4. Identify risks and issues to be raised at joint management reviews.

5. Ensure on-going communications between acquirer and developer technical personnel.

An entrance criteria for a technical review will require that an item to be reviewed is distributed to the group prior to the review meeting. Additionally a recorder will be assigned to record any issues requiring resolution stating action item assignee and due date, and decisions made within the authority of the technical review participants.

Various metrics are collected as part of technical reviews to help determine the effectiveness of the review process itself as well as the process steps that are used to produce the item being reviewed. These metrics, reported to the project manager, will include the amount of time spent by each person involved in the review, including preparation for the review.

6.1.2 Management Reviews

SQA's periodic management review of software project status, progress, problems, and risk will provide an independent assessment of project activities. SQA will provide the following information to management:

1. Compliance - Identification of the level of compliance of the project with established organizational and project processes.

2. Problem areas - identification of potential or actual project problem areas based on analysis of technical review results.

3. Risks - identification of risk based on participation and evaluation of project progress and trouble areas.

Because the SQA function is integral to the success of the project, SQA will freely communicate its results to senior management, project management and the project team. The method for reporting compliance, problem areas, and risks will be communicated in a documented report or memorandum. Compliance, problem areas, and risks will be followed-up and tracked to closure.

6.1.3 Process Audits

Software development processes are audited according to the tasks specified in Section 3.3, SQA Tasks, and performed in accordance with the software development schedule specified in the SDP.

6.1.4 Configuration Audits

6.1.4.1 Functional Configuration Audit

The Functional Configuration Audit (FCA) is held prior to the software delivery to compare the software as built (including its executable forms and available documentation) with the software requirements as stated in the baseline SRS. The purpose is to assure that the code addressed all, and only, the documented requirements and functional capabilities stated in the SRS. MIL-STD-973, Configuration Management, provides the guidelines for conducting an FCA. SQA will participate as a member of the FCA team with other FCA team members to be assigned by the project manager. SQA will assist in the preparation of the FCA findings. Any follow-up to the reported FCA finding will be monitored and tracked to closure.

6.1.4.2 Physical Configuration Audit

The Physical Configuration Audit (PCA) is held to verify that the software and its documentation are internally consistent and are ready for delivery. The purpose is to assure that the documentation to be delivered is consistent and correct describes the code. MIL-STD-973, Configuration Management, provides the guidelines for conducting an PCA. SQA will participate as a member of the PCA team with other PCA team members to be assigned by the project manager. SQA will assist in the preparation of the PCA findings. Any follow-up to the reported PCA finding will be monitored and tracked to closure.

This page intentionally left blank.

7. TEST

<Identify all other tests not included in verification and validation and state the methods used. Describe any testing techniques or methods which can be used to detect errors, to develop sets of test data, and to monitor computer system resources.>

[project name] testing activity includes unit level testing, integration testing (at Unit and CSCI/HWCI level), performance testing (CSCI Qualification Testing), and acceptance testing (System Qualification Testing). Figure 7-1 provides the Test Process Flow. SQA shall audit the testing activities as defined in the [project name] SDP, and shall ensure that the software and test documentation is subject to configuration management control. SQA shall witness the tests and verify that test results are recorded and evaluated. SQA shall coordinate the maintenance of STR logs with SCM and shall verify that software changes are controlled according to the SCM procedures. SQA shall witness all retesting resulting from STRs to verify the effectiveness of the correction.

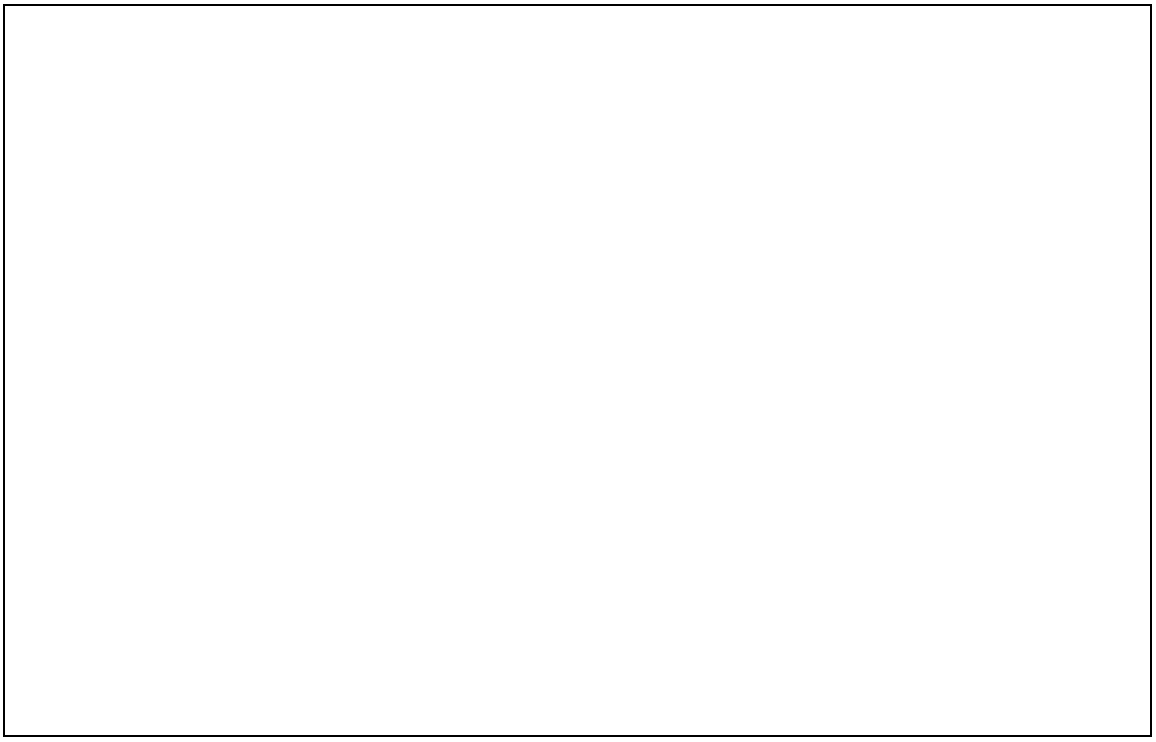

Table 7-1. Test Process Flow Diagram

This page intentionally left blank.

8. PROBLEM REPORTING AND CORRECTIVE ACTION

<Describe the practices and procedures to be followed for reporting, tracking, and resolving problems identified in both software items and the software development and maintenance process. State the specific organizational responsibilities concerned with their implementation.>

This section describes the reporting and control system used by SQA to record and analyze discrepancies and to monitor the implementation of corrective action. The forms utilized by SQA for reporting are the Process Audit Report, Software Trouble Report (STR), Software Tool Evaluation Report, Facilities Evaluation Report, and monthly SQA Status Report. Each of these forms and their uses are discussed in the following paragraph.

8.1 Process Audit Report

SQA reports the results of a process audit and provides recommendations, if necessary, using the Process Audit Report . The Process Audit Report is used to record that the process is (1) being followed correctly and working effectively, (2) being followed but is not working effectively, or (3) not being followed.

The Software Process Audit report is provided to the project manager. The project manager utilizes the report to provide insight into whether there is compliance with the development process and its effectiveness in meeting project goals. Where necessary and appropriate, the project manager may initiate enforcement activities or initiate change to the established processes using the approved procedures. Additionally, the Software Process Audit Report may be provided to senior management along with other project status information to guide senior management attention to identify and mitigate project risks at the organizational level.

Figure 8-1 provides the format of a Process Audit Report.

PROCESS AUDIT REPORT

TRACKING IDENTIFIER:_____

LEAD AUDITOR: DATE OF REPORT:

AUDIT TEAM:

PROJECT NAME:

DATE OF AUDIT:

PROCESS/PROCEDURE AUDITED:

AUDIT CHECKLIST USED: (Attach)

AUDIT FINDINGS: (Check one.)

_____ Process/Procedure Acceptable

_____ Process/Procedure Conditionally Acceptable
(Subject to satisfactory completion of action items listed below)
Conditions noted:

_____ Process/Procedure Unacceptable
(Subject to satisfactory completion of action items listed below)
Conditions noted:

ACTION ITEM (AI):
AI # TITLE: ASSIGNED TO: DUE DATE: COMP DATE:

CORRECTIVE ACTION:

DISPOSITION: APPROVE CANCEL DEFER
Project Manager: DATE:

AI CLOSURE:

SQA Sign-off: DATE:

(FILE COMPLETED FORM IN SQA EVALUATION RECORD.)

Figure 8-1. Process Audit Report

8.2 Problem/Change Report (P/CR)

Problems found in the software or software documentation which is under configuration management must be recorded by means of an P/CR regardless of how or by whom the problem was discovered. P/CRs generated by SQA shall be process in accordance with the [project name] SCMP. SQA shall analyze P/CRs for problem trends in an effort to prevent recurring discrepancies. SQA shall report the results of P/CR trend analyses along with suggestions for problem resolution and prevention. The format of the P/CR is found in the [project name] SCMP.

8.3 Software Tool Evaluation Report

Figure 8-2 provides the format for evaluating software tools.

8.4 Facilities Evaluation Report

Figure 8-3 provides the format for evaluating existing and planned [project name] facilities.

SOFTWARE TOOL EVALUATION

SQA:_____ DATE OF EVALUATION:_____

Software Tool Evaluated:

Methods or criteria used in the evaluation:

Evaluation Results:

Recommended Corrective Actions

Corrective Action Taken

Figure 8-2. Software Tool Evaluation

PROJECT FACILITIES EVALUATION

SQA:_____ DATE OF EVALUATION:_____

Facility Evaluated (Equipment, User/Test/Library Space):

Methods or criteria used in the evaluation:

Evaluation Results:

Recommended Corrective Actions

Corrective Action Taken

Figure 8-3. Project Facilities Evaluation

This page intentionally left blank.

9. TOOLS, TECHNIQUES, AND METHODOLOGIES

<Identify the special software tools, techniques, and methodologies that support SQA, state their purposes, and describe their use.

Tools - SQA software tools include, but are not limited to, operating system utilities, debugging aids, documentation aids, checklists, structuring preprocessors, file comparators, structure analyzers, code analyzers, standards auditors, simulators, execution analyzers, performance monitors, statistical analysis packages, software development folder/files, software traceability matrices, test drivers, test case generators, static or dynamic test tools, and information engineering CASE tools.

Techniques - techniques include review of the use of standards, software inspections, requirements tracing, requirements and design verification, reliability measurements and assessments, and rigorous or formal logic analysis.

Methodologies - methodologies are integrated set of the above tools and techniques. The methodologies should be well-documented for accomplishing the task or activity and provide a description of the process to be used.>

Where applicable, SQA will use SEPO organizational processes and tailor the processes specific to the project.

This page intentionally left blank.

10. CODE CONTROL

<Define the methods and facilities used to maintain, store, secure and document controlled versions of the identified software during all phases of the software life cycle. This may be implemented in conjunction with a computer program library. This may be provided as a part of the SCMP. If so, an appropriate reference should be made.>

Code control includes:

(1) identifying, labeling, and cataloging the software to be controlled,
(2) identifying the physical location of the software under control,
(3) identifying the location, maintenance, and use of backup copies,
(4) distributing copies of the code,
(5) identifying the documentation that is affected by a change,
(6) establishing a new version, and
(7) user access to the code.

[project name] uses [identify CM Code Control Software] for code control. The code control method is described in the [project name SCMP]. SQA will conduct ongoing evaluations of the code control process to ensure that the process of controlling the code is effective and in compliance with the [project name] SCMP.

This page intentionally left blank.

11. MEDIA CONTROL

<State the methods and facilities to be used to identify the media for each computer product and the documentation required to store the media, including the copy and restore process, and projects computer program physical media from unauthorized access or inadvertent damage or degradation during all phases of the software life cycle. This may be provided as a part of the SCMP. If so, an appropriate reference should be made.>

Media control includes:

(1) regularly scheduled backup of the media,
(2) labeled and inventoried media filed in a storage area in accordance with security requirements and in a controlled environment that prevents degradation or damage to the media, and
(3) adequate protection from unauthorized access.

The software media control methods and facilities is described in the [project name] SCMP. SQA will conduct ongoing evaluations of the software media control process to ensure that the process of controlling the software media is effective and in compliance with the [project name] SCMP.

This page intentionally left blank.

12. SUPPLIER CONTROL

<State the provisions for assuring that software provided by suppliers meets established requirements.>

Prior to any purchase of software to support the development effort, SQA and project members will define and provide complete requirements to the supplier/vendor. The Software Tool Evaluation process will be followed. Part of the evaluation process will require the supplier/vendor to describe then technical support, handling of user questions and problems, and software product upgrades.

<In some cases project do no foresee purchasing software. If that's the case, the following paragraphs may apply.>

All supplier software has been operationally tested in the target system. No future supplier software is planned.

This page intentionally left blank.

13. RECORDS COLLECTION, MAINTENANCE AND RETENTION

<Identify the SQA documentation to be retained, state the methods and facilities to be used to assemble, safeguard, and maintain this documentation, and designate the retention period.>

SQA activities are documented by records and reports which provide a history of product quality throughout the software life cycle. Metric data collected will be reviewed for trends and process improvement. All SQA records will be collected and maintained in the SDL or archival storage for the life cycle of the product or a minimum of [state number of years].

This page intentionally left blank.

14. TRAINING

<Identify the training activities necessary to meet the needs of the SQAP.>

Table 14-1 provides a matrix that identifies the required skills to perform SQA tasks to implement this [project name] SQAP. The training schedule will be compatible with the project schedule. In some cases, training will be conducted as on-the-job training.

Figure 14-1. SQA Training Matrix

TASK	SKILL REQUIREMENTS
Code Reviews	Source Language, Peer Reviews
Documentation Reviews	Software Development and Documentation standards and guidelines, Peer Reviews
Process Audits	Software Development Life Cycle Processes, Audit techniques
Testing	Testing Methodologies
SQA Management	Project Management
Metrics	Data Collection and Analysis
Problem reporting and correction action	Configuration Management
Tools	Vendor supplied training
Code, Media, and Supplier Control	Configuration Management
CMM KPA Audits	CMM, CMM Based Appraisals
SPI	SPI process
Risk Management and Analysis	Risk Management Process

This page intentionally left blank.

15. RISK MANAGEMENT

<Specify the methods and procedures employed to identify, assess, monitor, and control areas of risk arising during the portion of the software life cycle covered by the SQAP.>

The [project name] has developed a risk management plan as identified in [document name]. SQA will review and evaluate the technical risk analysis and any risk reduction plan.

This page intentionally left blank.

APPENDIX A

LIST OF ACRONYMS

AI	Action Item
CDR	Critical Design Review
CMM	Capability Maturity Model
CMU	Carnegie-Mellon University
CRLCMP	Computer Resource Life Cycle Management Plan
CSCI	Computer Software Configuration Item
DBDD	Data Base Design Description
DCR	Document Change Request
DID	Data Item Description
DOD	Department of Defense
FCA	Functional Configuration Audit
FQR	Formal Qualification Review
HB	Handbook
HWCI	Hardware Configuration Item
IDD	Interface Design Description
IEEE	Institute of Electrical and Electronics Engineers
IRS	Interface Requirements Specification
IV&V	Independent Verification and Validation
KPA	Key Process Area
MIL	Military
NDS	Non-Developmental Software
OCD	Operational Concept Document
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PP&O	Project Planning and Oversight
PRR	Product Readiness Review
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SDD	Software Design Document
SDF	Software Development File
SDP	Software Development Plan
SDR	System Design Review
SEI	Software Engineering Institute
SEI	Software Engineering Institute
SEPO	Software Engineering Process Office
SPAWARSYSCEN	Space and Naval Warfare Systems Center
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SRR	System Requirements Review
SRS	Software Requirements Specification

SSDD	System/Subsystem Design Description
SSR	Software Specification Review
SSS	System/Subsystem Specification
STD	Standard
STR	Software Trouble Report
SVD	Software Version Description
SW-CMM	Software Capability Maturity Model
TR	Technical Report
TRR	Test Readiness Review
UDF	Unit Development Folder
[project name]	